DOCUMENT RESUME

ED 384 656 TM 023 949

AUTHOR Stocking, Martha L.; And Others
TITLE An Experiment in the Application of an Automated Item Selection Method to Real Data.
INSTITUTION Educational Testing Service, Princeton, N.J.
REPORT NO ETS-RR-91-64
PUB DATE Dec 91
NOTE 26p.; Version of a paper presented at the Annual Meeting of the National Council on Measurement in Education (Boston, MA, April 16-20, 1990).
PUB TYPE Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Algorithms; *Automation; Coding; *Computer Assisted Testing; Item Banks; *Selection; *Test Construction; Test Items
IDENTIFIERS Test Specifications

ABSTRACT

A previously developed method of automatically selecting items for inclusion in a test subject to constraints on item content and statistical properties is applied to real data. Two tests are first assembled by experts in test construction who normally assemble such tests on a routine basis. Using the same pool of items and constraints articulated by test construction experts the same two tests are reassembled automatically. The manual and automatic assemblies are compared by test specialists who were not involved in the original manual assembly. Based on this experiment, barriers to future progress in the improvement of automatic test assembly methods seem not to be in the development of different algorithms, nor in the improvement of computer time and cost. Rather, the focus of future improvements in the process of automatic test assembly lies more in the direction of complete specifications of constraints on item selection and detailed coding of item properties. (Contains 14 references, 2 tables, and 2 figures.) (Author)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

RR-91-64

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
H. I. BRAUN
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

ED 384 656

RESEARCH REPORT

AN EXPERIMENT IN THE APPLICATION OF AN AUTOMATED ITEM SELECTION METHOD TO REAL DATA

Martha L. Stocking
Len Swanson
Mari Pearlman

BEST COPY AVAILABLE

Educational Testing Service
Princeton, New Jersey
December 1991

TM023949

AN EXPERIMENT IN THE APPLICATION OF AN AUTOMATED ITEM SELECTION METHOD TO REAL DATA*

Martha L. Stocking
Len Swanson
Mari Pearlman

Educational Testing Service
Princeton, New Jersey 08541

October 1991

*This work was supported by Educational Testing Service through the Program Research Planning Council. Part of this paper was presented at the annual meeting of the National Council on Measurement in Education, Boston, 1990.

Copyright © 1991. Educational Testing Service. All rights reserved.

AN EXPERIMENT IN THE APPLICATION OF AN

AUTOMATED ITEM SELECTION METHOD TO REAL DATA

Abstract

A previously developed method of automatically selecting items for inclusion in a test subject to constraints on item content and statistical properties is applied to real data. Two tests are first assembled by experts in test construction who normally assemble such tests on a routine basis. Using the same pool of items and constraints articulated by test construction experts the same two tests are reassembled automatically. The manual and automatic assemblies are compared by test specialists who were not involved in the original manual assembly. Based on this experiment, barriers to future progress in the improvement of automatic test assembly methods seem not to be in the development of different algorithms, nor in the improvement of computer time and cost. Rather, the focus of future improvements in the process of automatic test assembly lies more in the direction of complete specifications of constraints on item selection and detailed coding of item properties.

Key words: Test Assembly, Test Construction, Mathematical Programming, Heuristic Algorithms, Test Design

AN EXPERIMENT IN THE APPLICATION OF AN AUTOMATED ITEM SELECTION METHOD TO REAL DATA

Introduction

The process of test construction is time consuming and expensive. Every year public and private testing organizations spend millions of dollars writing, editing, and otherwise building items and then assembling these items into test forms for some measurement purpose. Until recently, the process of test construction or test assembly has been virtually unassisted by modern psychometrics.

A practical application of Item Response Theory (IRT) suggested by Lord (1980) and Birnbaum (1968) is to use IRT in the test assembly process. But the methods they suggest have little or no practical (as opposed to measurement) advantage over methods using conventional statistics without additional assistance from modern computers. In the last ten years, many test assembly paradigms employing a combination of IRT, modern computers, and mathematical programming methods or heuristic methods have been proposed in the literature. Most of the studies reported on these new paradigms have used artificial data in research settings with few constraints on the item selection. Exemplars of such studies include Thunissen (1985), Thunissen (1986), Baker (1988), van der Linden (1987), van der Linden and Boekkooi-Timminga (1989), and de Gruijter (1990). These research settings may be appropriate for the preliminary investigation of new test assembly paradigms, but may not resemble very closely problems encountered when assembling tests from real item pools with more numerous constraints on item selection. Some studies have used quasi-realistic data with more (and perhaps more realistic)

constraints, for example, Ackerman (1989) and Stocking, Swanson, and Pearlman (1991). Here the practicality of various paradigms becomes more obvious.

This paper presents an experiment with the Stocking, Swanson, and Pearlman (1991) heuristic item selection algorithm using real data and real constraints actually employed in the manual test assembly process for a particular test. In this experiment, a test form is first assembled manually using the test construction practices currently in place. Then the test form is reassembled from the same item pool using the automatic item selection algorithm. The two forms are compared in terms of item overlap, statistical properties, and content properties and the results are discussed.

Theoretical Framework

Most of the recently published test construction paradigms depend upon the use of IRT. Within this context, some aspect of the items to be selected is optimized subject to constraints on other item properties. Table 3 of van der Linden and Boekkooi-Timminga (1989) lists the functions optimized in a number of useful models. The constraints may incorporate characteristics of items, such as content or type, that are important in test construction. The constraints may also incorporate desirable properties of the resultant test information function. Some paradigms invoke a formal optimization strategy familiar from the field of mathematical programming, for example van der Linden (1987). This guarantees optimal solutions but may sacrifice practicality. Others invoke more informal heuristic optimization strategies, such as Adema (1988), Adema (1989) or Ackerman (1989), which may sacrifice global optimality for the sake of practical gains.

The automated item selection algorithm (Stocking, Swanson, and Pearlman, 1991) is of the heuristic type. The rationale for this algorithm is as follows: Typical test assembly is less concerned with optimizing some function of the items selected (for example, maximizing test information or minimizing test length) or even meeting all of the constraints of interest, than it is with coming "as close as possible" to all constraints simultaneously. Phrased another way, when confronted with a large number of constraints it is often better to miss on two or three of them, but come very close, than meet all but one constraint but miss that one by a large margin.

Thus "constraints", including statistical constraints, are thought of as more "desired properties" than true constraints. This approach recognizes the possibility of constructing a test that may lack all of the desired properties, but emphasizes the minimization of aggregate failures. If a constraint is not satisfied, the heuristic allows consideration of the extent to which it is violated. Moreover, the heuristic provides for the possibility that not all constraints are equally important to the test constructor by incorporating explicit weights as part of the modeling of constraints.

Thus the goal of this heuristic can be stated very simply: *minimize the weighted sum of deviations from the constraints*. The constraints are formulated as bounds on the number of items having specified properties. The constraints need not, and in general will not, divide the item pool into mutually exclusive subsets. Rather each item can have many different features satisfying many different constraints. This is a very general formulation which can incorporate constraints related to test information functions in the context of IRT but can also function equally well in the non-IRT context where statistical constraints might be formulated in terms of conventional item

difficulty and discrimination indices. The formulation works on any set of linear constraints, including constraints which treat only non-statistical properties of items. The same approach can also easily be formulated to incorporate nonlinear constraints.

In the mathematical expression of models in this framework, decision variables x_i, $i = 1,\ldots,N$, are defined for each item in an N-item pool. These decision variables take on the values of $x_i = 0$ if the item is excluded from the test being assembled and $x_i = 1$ if the item is included in the test. The complete model for binary linear constraints in an IRT context is expressed mathematically as follows:

Let $i = 1,\ldots,N$ index the items in the item pool,

θ_k, $k = 1,\ldots,K$ be the K values of θ at which the test information function, or item information functions, are evaluated. Then

minimize

$$\sum_{r=1}^{2K+2m} w_r D_r \ , \tag{1}$$

subject to

$$I_L(\theta_k) - \sum_{i=1}^{N} I_i(\theta_k) x_i = d_k \leq 0, \quad k = 1,\ldots,K \ , \tag{2}$$

$$\sum_{i=1}^{N} I_i(\theta_k) x_i - I_U(\theta_k) = d_{K+k,} \leq 0 \ , \quad k = 1,\ldots,K \ , \tag{3}$$

$$L_j - \sum_{i=1}^{N} a_{ij} x_i = d_{2K+j} \leq 0 \ , \quad j = 1,\ldots,m \ , \tag{4}$$

$$\sum_{i=1}^{N} a_{ij}x_i - U_j \equiv d_{2K+m+j} \leq 0 \ , \quad j = 1,\ldots,m \ , \tag{5}$$

$$\sum_{i=1}^{N} x_i = n \ , \tag{6}$$

and

$$x_i \in (0,1), \ i = 1,\ldots,N \ , \tag{7}$$

where $D_r \equiv \{ d_r \text{ if } d_r > 0; \ 0 \text{ otherwise} \}$ for $r = 1,\ldots,2K + 2m$; m is the number of linear constraints; $2K + 2m$ is the total number of constraints to be considered; the w_r are the weights to be applied to each constraint; the $I_L(\theta_k)$ and $I_U(\theta_k)$ are the set of lower and upper target information values, respectively; the L_j and U_j are the lower and upper bounds on the other constraints, respectively; and the a_{ij} are 1 if item i has property j, otherwise $a_{ij} = 0$.

The D's are a measure of the magnitude of the constraint violations, and are zero as soon as the relevant constraint is met. This implies that there may be many solutions to the optimization problem, that is, many sets of items that can be selected that satisfy all of the constraints; each of these solutions is equally satisfactory from the test assembler's viewpoint.

This formal expression of the model demonstrates that it can be viewed as a mixed integer linear programming problem (MILP) (see, for example, Nemhauser and Wolsey, 1988). However, standard methods of obtaining solutions

to this optimization problem are not practical in the context of test assembly because they cannot be easily applied to such real-world complications as item sets. An item set is a group of items associated with common stimulus material, typified by a reading passage followed by a set of questions about that passage, or a table or graph with a set of questions about that table or graph. Test assemblers often have constraints on the sets themselves ("select no more than two science-related passages"), as well as non-mutually exclusive constraints on the individual items within the set, and often not all of the items within a set are to be included in the test. There is no easy way of incorporating sets in the linear optimization paradigm since inclusion of an arbitrary subset of a set within a test is implied by inclusion of any one of the 2^n-1 possible combinations of n items within that set. The heuristic algorithm developed by Stocking, Swanson, and Pearlman, treats this and other practical issues and falls into the class of "greedy" heuristic algorithms (Nemhauser and Wolsey, Chapter II.5). The algorithm selects items in such a way that during the search for the solution the expected sum of deviations from constraints is never increased, but either remains the same or is reduced.

The Experiment

The test selected for this experiment consists of a verbal measure and a quantitative measure. The verbal measure is 70 items long and consists of three sections: Reading Comprehension (23 items),. Sentence Correction (an indirect measure of writing that is 27 items long), and Critical Reasoning (a measure of verbal reasoning that is 20 items long). The quantitative measure consists of two parallel Problem Solving sections, each 20 items long, that

contain a balance of items on arithmetic, algebra, and geometry, and a third section, 25 items long, that contains Data Sufficiency questions.

The pool from which the verbal test is constructed for this experiment consists of 1,538 items, including 55 reading comprehension passages with 11 to 12 items associated with each passage. The pool from which the quantitative test is constructed consists of 853 items, including 11 quantitative stimuli with 4 to 5 items associated with each stimulus. Both pools have been (independently) calibrated using the 3-parameter logistic (3PL) item response function model (see Lord, 1980) and the computer program LOGIST (Wingersky, 1983). Information about the items, including their estimated item parameters and content classifications, is stored in an electronic PC-based item-banking system.

The Manual Test Assembly

In the current manual test assembly process for these measures, test assemblers work with target test information functions. The targets for each measure are based on the attributes of previous test editions. Since the new test is required to be as parallel as possible to previous editions, both in terms of content and in terms of statistical properties, there are two targets for each measure representing the desirable range in which the resultant test information function must lie. If the information function for the test under construction lies extensively outside this range, it is considered to be less than optimally parallel to previous editions. The target information functions for each measure are shown as the lower and upper curves in Figures 1 and 2.

For the verbal measure, current practice is to assemble the three sections in a serial fashion, starting with Reading Comprehension, then

Critical Reasoning, and finally Sentence Correction. This practice developed over the history of this test and was based on the observation that the assembly of the Reading Comprehension section is subject to the most constraints (because of the passages) and has the least extensive item pool. The Critical Reasoning section has only a moderately extensive pool, while the Sentence Correction section has a reasonably extensive pool. Thus it is always likely that any measurement deficits (in terms of test information) that exist after the assembly of the first two sections may be compensated for in the assembly of the final section.

During the actual assembly, a test construction specialist chooses an item or a group of items to include in the test, then checks the resultant partial information function against the targets. This process is repeated, in very much the fashion suggested by Lord (1980), until the entire section is complete. The completed section is then turned over to the individual responsible for assembling the next section.

The assembly of the quantitative measure is a simpler process since two separate sections are constructed simultaneously to be parallel, and the third is then added to complete the total test information function. Items with approximately the same content and statistical properties are chosen in pairs for the two parallel sections using frequency distributions of item difficulties and discriminations. When the third section, data sufficiency, is assembled using a frequency distribution, the three are combined and their total information function is compared to the target functions.

The statistical results of the manual assemblies for the two measures are shown as dotted lines in Figures 1 and 2 with the companion target test information functions. As would be expected, the measurement requirements of

the resultant tests are predominantly satisfied. In addition, all of the non-statistical specifications that control the selection of items have been met.

The Automatic Assembly

All properties of items considered relevant to test assembly are coded in the PC-based item-banking system. Close observation and interaction with test construction experts led to the realization that tests are assembled using not only the unique specifications for a particular test, but also using more general specifications that apply to a broader class of tests of which a particular test is a member. These additional specifications generally incorporate what are considered to be "good test construction practices" for assembling multiple choice tests of this kind and are automatically taken into account by experienced test assemblers. However, for any automatic method of test assembly to be effective, these types of more general constraints must be made explicit.

Tables 1 and 2 display the complete list of (generally non-mutually exclusive) non-statistical item features determined to be relevant to the assembly of the verbal and quantitative measures, respectively. The verbal measure is assembled using 43 item features; the quantitative measure is assembled using 144 features. The tables are intentionally displayed in reduced size because a detailed understanding of the table entries is not necessary. Our purpose here is not to provide a complete discussion and rationale for each feature, but rather to give a general overview of the number and nature of all non-statistical features controlling the process of test assembly. In the tables, each feature is given an abbreviated name that indicates what item properties are relevant for the feature. This may be a single item feature as in, for example, the first line in Table 1, labeled

MAINID, which specifies items that are classified as questioning some aspect of the main idea of a reading passage. Multiple item features may be considered together as in, for example, the first line in Table 2, labeled DS4AB which specifies data sufficiency items coded in field 4 of the database as involving absolute values.

Each item feature has associated with it a lower and upper bound on the number of items with these features that may appear in a test. These lower and upper bounds may, of course, be equal. If the bounds are not equal, then two constraints are added to the optimization problem as specified in equations 4 and 5. If the bounds are equal, then only a single constraint is added to the optimization problem as equations 4 and 5 become equivalent.

Constraints generated by features marked with an asterisk in the Tables are specifications mandated by the specific purpose of this particular test. The others, often more detailed, are determined by commonly accepted test construction practices. As can be seen from these tables, the verbal measure has a preponderance of test specific features (33 out of 43) while the quantitative measure has a preponderance of more general features (130 out of 144).

The total number of non-statistical constraints for the optimization problem for the verbal measure is 75, (2 times the number of features with unequal lower and upper bounds plus the number of features with equal lower and upper bounds). The total number of non-statistical constraints for the quantitative measure is 282.

The automatic assemblies were also subject to statistical constraints in terms of the resultant test information function, as were the manual assemblies previously described. The information functions were required to

lie at or above the lower target test information functions and at or below the upper target test information functions at selected values on the ability metric. The number of ability levels at which these constraints are enforced determines the number of constraints added to the automatic test assembly problem. The actual ability levels used for the final test assembly may be chosen based on some prior knowledge of the strengths and weaknesses of the item pool as a whole, or through an iterative process where intermediate solutions are seen to be less than satisfactory. For the verbal measure, the constraints on the test information function were specified at 7 ability levels (theta = -2.1, -1.5, -.9, 0., .9, 1.5, 2.1), thus adding 14 constraints to the verbal solution. The total number of items to be selected serves as a final constraint, bringing the total number of constraints for the verbal solution to 90. For the quantitative measure, the constraints were specified at 11 ability levels (theta = -2.4. -1.8, -1.5, -1.2, -.9, -.6, -.3, 0., .9, 1.8, 2.4) adding 22 constraints for a total of 305 constraints on the solution, including a constraint on the total number of items in the test.

The Results

The automated item selection algorithm was implemented on a Compaq 386 with a 80387 math coprocessor, running at 20 megahertz. All of the constraints for both the verbal and quantitative measures were weighted equally at 1.0. It took 10 minutes to assemble a 70-item verbal measure from a pool of 1,538 items, subject to 90 constraints, and 8.5 minutes to assembly a 65-item quantitative measure from a pool of 853 items subject to 305 constraints. These are total times that include both the time required for the algorithm and the time required to retrieve relevant data from the data base. The latter components of the total time are clearly implementation-

based, and while they are important in terms of practical applications, they are less important when comparing properties of algorithms. The amount of time directly attributable to the algorithm was 2.6 minutes or 155 seconds (out of 10 minutes) for the assembly of the verbal measure and 2.4 minutes or 145 seconds (out of 8.5 minutes) for the assembly of the quantitative measure. The total times seem acceptable given that the time to assemble the two measures manually from these pools is approximately three person-days.

The results of the automatic assembly in terms of test information are shown as dashed lines in Figures 1 and 2. The test information functions lie predominantly within the lower and upper target test information functions within the ability range of interest. Although different from those for the manual test assembly process shown in the same Figures, these results are clearly acceptable. These statistical results were obtained with complete satisfaction of all non-statistical constraints specified for the algorithm and listed in Tables 1 and 2.

When the tests assembled automatically were compared with the tests assembled by test specialists, there was little overlap of the actual items selected. For the entire verbal measure, only 12 out of 70 items (all associated with a single reading passage) were selected by both methods whereas for the quantitative measure there were 7 out of 65 items in common. Clearly the large item pools are rich enough to support the assembly of a number of parallel editions of the measures without compromising either statistical or content constraints.

As a final check on the forms assembled automatically, each form was given to test specialists for review. These test specialists were also ones

who normally assemble the final forms manually but were not involved the manual assembly for this experiment.

The comments made by the test specialists tended to fall into two distinct categories. One category concerned items that individual test assemblers considered to be of poor substantive quality for a variety of reasons. Items with poor statistical properties were, for the most part, never included in the pools available for this experiment. The elimination of the problematic items identified on a substantive basis by test specialists could be handled in a similar manner. Eliminating them from the pool guarantees that they are never selected for inclusion in a test assembled by the heuristic. Incorporating them appropriately in the constraints guarantees that they are only selected when absolutely necessary. The former solution requires no additional item coding; the latter solution requires additional item coding but is probably closer to what test assemblers do in practice.

A second category of comments from test specialists concerned the assembled tests as a whole and the classification of items represented by the assemblies. Some subcategories of classifications of items were not previously identified as important and therefore were not part of the item coding or the specification of constraints. In addition, some items were identified previously as having only a single property (for example, a verbal item on the abolitionist movement) when they should have been identified as having two properties (for example, the role of women in the abolitionist movement). Thus this exercise has proved useful in eliciting from test construction specialists a more complete list of unique specifications for this test. A more detailed classification and coding of items as well as the

specification of additional constraints are required to satisfy these additional specifications.

Discussion and Conclusions

The implementation of the automated item selection algorithm was technically successful. All constraints for both measures were satisfied in a reasonable amount of computer time. There were two areas of possible improvement in the process. First, a more detailed and careful consideration of items included in the item pools is clearly necessary. Either substantively problematic items should be removed from the pool, or they should be included along with the appropriate constraints to the algorithm so that they are chosen only if no alternative is available.

Second, it seems clear that we have not yet achieved the complete specification of all constraints. The process of developing such complete specifications should most likely be iterative and interactive. With the knowledge gained so far, we can develop additional constraints and try again. It seems likely that after a few more iterations we will have a set of constraints that is more satisfactory.

With this expanded set of constraints, the automatic assembly of tests should result in very good tests that may be more parallel to each other than those constructed through a manual process. However, it seems unlikely that any automatic process can ever produce a test that can be considered finished without any human intervention. The automated item selection algorithm tried out here, as well as other test assembly algorithms in the literature, can probably never capture that part of test assembly that is clearly more of an "art" rather than a "science."

Subsequent to the experiment described here, the heuristic algorithm has been successfully applied to the assembly of 14 other tests or test sections from item pools ranging in size from around 300 items to over 5,000 items and subject to constraints that numbered anywhere from 30 to over 500. A number of these tests or test sections used conventional rather than IRT-based item statistics in the assembly process. Nothing that we have learned from this experiment was contradicted by these more extensive applications.

Based on this experiment and the more comprehensive applications, barriers to future progress in the improvement of automatic test assembly methods seem not to be in the development of different algorithms, nor in the improvement of computer time and cost. Rather, the focus of future improvements in the process of automatic test assembly lies more in the directions of complete specifications of constraints on item selection and the detailed coding of item properties.

References

Ackerman, T. (1989, March). An alternative methodology for creating parallel test forms using the IRT information function. Paper presented at the 1989 NCME annual meeting, San Francisco.

Adema, J. J. (1988). A note on solving large-scale zero-one programming problems (Research Report 88-4). Enschede: Department of Education, University of Twente.

Adema, J. J. (1989). Implementations of the Branch and Bound method for test construction problems (Research Report 89-6). Enschede, The Netherlands: Department of Education, University of Twente.

Baker, F. B., Cohen, A. S., & Barmish, B. R. (1988). Item characteristics of tests constructed by linear programming. Applied Psychological Measurement, 12, 189-199.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord and M. R. Novick, Statistical theories of mental test scores (pp. 395-479). Reading, MA: Addison-Wesley.

de Gruijter, D. N. M. (1990). Test construction by means of linear programming. Applied Psychological Measurement, 14, 175-181.

Lord, F. M. (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Erlbaum.

Nemhauser, G. L., & Wolsey, L. A. (1988). Integer and combinatorial optimization. New York, N. Y.: John Wiley & Sons.

Stocking, M. L., Swanson, L., & Pearlman, M. (1991). Automated item selection using item response theory. (Research Report 91-9). Princeton, N J.: Educational Testing Service.

Theunissen, T. J. J. M. (1985). Binary programming and test design. Psychometrika, 50, 411-420.

Theunissen, T. J. J. M. (1986). Some applications of optimization algorithms in test design and adaptive testing. Applied Psychological Measurement, 10, 381-389.

van der Linden, W. J. (1987). Automated test construction using minimax programming. In W. J. van der Linden (Ed.), IRT-based test construction. Enschede, The Netherlands: Department of Education, University of Twente.

van der Linden, W. J., & Boekkooi-Timminga, E. (1989). A maximin model for test design with practical constraints. Psychometrika, 54, 237-248.

Wingersky, M. S. (1983). LOGIST: A program for computing maximum likelihood procedures for logistic test models. In R. K. Hambleton (Ed.), Applications of item response theory. Vancouver, BC: Educational Research Institute of British Columbia.

Table 1: The non-statistical features controlling the selection of items and sets for the verbal measure. Those features marked with an asterisk are specific to this measure.[1]

Name	Lower Bound	Upper Bound	Number Chosen	Name	Lower Bound	Upper Bound	Number Chosen
*MAINID	2	3	3	*OPSSG	1	1	1
*SUPID	6	8	8	*BPSSG	1	1	1
*INFER	6	8	7	*HPSSG	0	1	0
*APPL	2	4	2	*AGREE	1	3	3
*EVAL	2	4	2	*CSTGRA	1	3	1
*STYLE	0	2	1	*CSTNE	3	6	3
* SS	7	9	7	*DICTIO	1	3	1
* PS	7	9	7	*IDIOM	2	5	5
* BU	7	9	9	*LOGPRE	5	9	9
*SSPSSG	1	1	1	*PARALL	1	3	2
*BSPSSG	0	0	0	*VRBFRM	2	4	3
*PSPSSG	1	1	1	AGREE4	2	4	3
*BUPSSG	1	1	1	CSTGRA4	2	4	2
GENDER	1	1	1	CSTNE4	2	5	4
*BUSINS	8	15	11	DICTIO4	0	2	1
CRDIFC	0	0	0	IDIOM4	2	5	5
*CRTI	6	12	9	LOGPRE4	5	9	8
*CRTII	6	12	8	PARALL4	1	3	1
*CRTIII	1	6	3	VRBFRM4	1	5	3
*HUMAN	0	2	0	*RCRAN	23	23	23
*OTHER	5	10	9	*SCRAN	27	27	27
*CRRAN	20	20	20				

[1]Abbreviated feature names containing the characters PSSG are features of reading passages.

Table 2: The non-statistical features controlling the selection of items and sets for the quantitative measure. Those features marked with an asterisk are specific to this measure.[1]

Name	LB	UB	NC	Name	LB	UB	NC	Name	LB	UB	NC
DS4AB	0	2	0	DSAG	1	2	1	PSALP	1	2	1
DS4AM	0	2	0	DSALR	1	2	1	PSALR	1	2	1
DS4AN	0	2	0	DSALP	1	2	1	PSAOR	1	2	1
DS4AV	0	2	2	*DSALG	6	7	7	PSAOP	1	2	1
DS4BB	0	2	0	*DSARP	6	7	7	*PSAPU	6	8	6
DS4BO	0	2	0	*DSARR	4	5	4	*PSARE	6	8	6
DS4C3	0	2	0	*DSGP	2	2	2	*PSARP	10	12	10
DS4CA	0	2	0	*DSGR	1	1	1	*PSARR	12	14	14
DS4CC	0	2	0	DSKEYA	2	7	4	PSSEX	0	6	0
DS4CD	0	2	1	DSKEYB	2	7	4	*PSGP	2	2	2
DS4CF	0	2	1	DSKEYC	2	7	6	*PSGR	2	2	2
DS4CG	0	2	0	DSKEYD	2	7	6	PSROMA	0	2	1
DS4CI	0	2	1	DSKEYE	2	7	3	PS4ML	0	4	0
DS4CO	0	2	0	PS4AB	0	4	0	PS4MS	0	4	1
DS4CP	1	2	2	PS4AM	0	4	1	PS4OE	0	4	0
DS4CR	0	2	1	PS4AN	0	4	2	PS4OR	0	4	3
DS4E2	0	2	1	PS4AV	0	4	0	PS4PC	0	4	1
DS4ES	0	2	0	PS4BB	0	4	0	PF4PF	0	4	1
DS4FF	0	2	0	PS4BO	0	4	0	PF4PG	0	4	0
DS4FM	0	2	2	PS4C3	0	4	0	PS4PL	0	4	1
DS4GR	0	2	0	PS4CA	0	4	1	PS4PP	0	4	0
DS4IF	0	2	0	PS4CD	0	4	3	PS4PV	0	4	0
DS4IN	0	2	0	PS4CF	0	4	1	PS4PY	0	4	1
DS4L3	0	2	0	PS4CG	0	4	1	PS4QA	0	4	0
DS4LL	0	2	0	PS4CI	0	4	3	PS4QE	0	4	0
DS4LO	0	2	0	PS4CO	0	4	0	PS4QP	0	4	0
DS4LS	0	2	0	PS4CP	0	4	3	PS4R3	0	4	0
DS4MC	0	2	0	PS4CR	0	4	1	PS4RE	0	4	0
DS4MG	0	2	0	PS4E2	0	4	0	PS4RP	0	4	2
DS4ML	0	2	0	PS4ES	0	4	0	PS4RT	0	4	2
DS4MS	0	2	0	PS4FF	0	4	0	PS4S3	0	4	0
DS4PY	0	2	0	PS4FM	0	4	2	PS4SP	0	4	0
DS4QA	0	2	1	PS4GR	0	4	2	*PSRAN	40	40	40
DS4QE	0	2	0	PS4IF	0	4	0	*DSRAN	25	25	25
DS4QP	0	2	0	PS4IN	0	4	0	DS4OE	0	4	0
DS4R3	0	2	1	PS4L3	0	4	0	DS4ORE	0	4	3
DS4RE	0	2	0	PS4LL	0	4	0	DS4PC	0	4	0
DS4RP	0	2	2	PS4LO	0	4	0	DS4PF	0	4	0
DS4RT	0	2	1	PS4LS	0	4	0	DS4PG	0	4	0
DS4S3	0	2	0	PS4MC	0	4	0	DS4PL	0	4	0
DS4SP	0	2	1	*DALRA	4	5	4	DS4PP	0	4	1
DS4SS	0	2	0	PS4MG	0	4	0	DS4PV	0	4	0
DS4ST	0	2	0	PS4SS	0	4	1	DS4WK	0	2	0
DS4TA	0	2	1	PS4ST	0	4	1	DS4XP	0	2	1
DS4TO	0	2	0	PS4TA	0	4	0	DSALQ	1	2	1
DS4TP	0	2	0	PS4TO	0	4	0	PS4TT	0	4	0
DS4TR	0	2	0	PS4TP	0	4	0	PS4WK	0	4	1
DS4TT	0	2	0	PS4TR	0	4	1	PS4XP	0	4	1

[1]LB stands for Lower Bound; UB stands for Upper Bound; NC stands for Number Chosen.

BEST COPY AVAILABLE

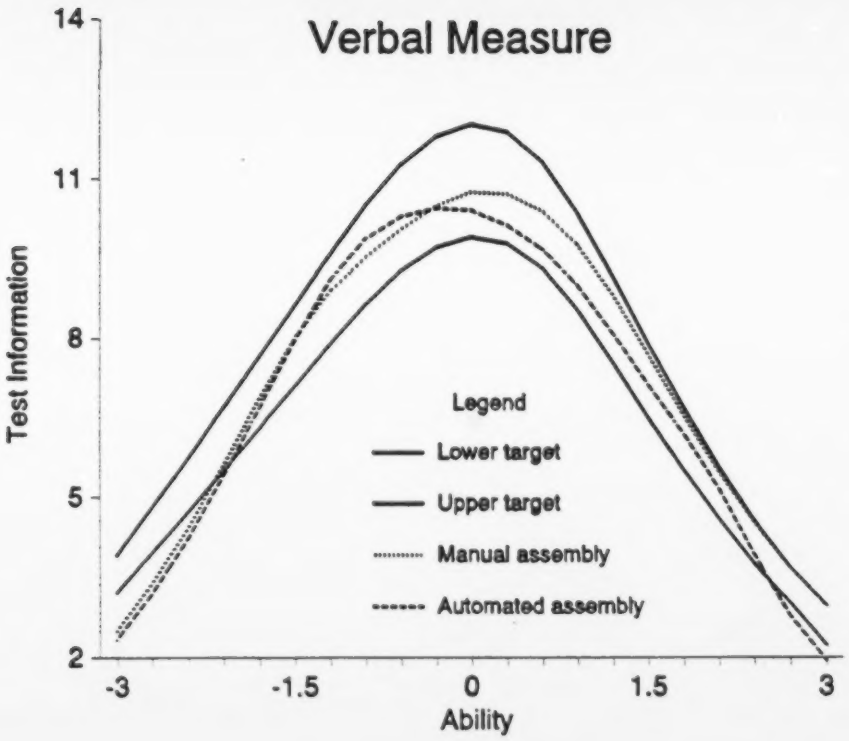

Figure 1: Upper and lower target test information functions and the resultant test information functions for the manual and automatic assemblies of the verbal measure.

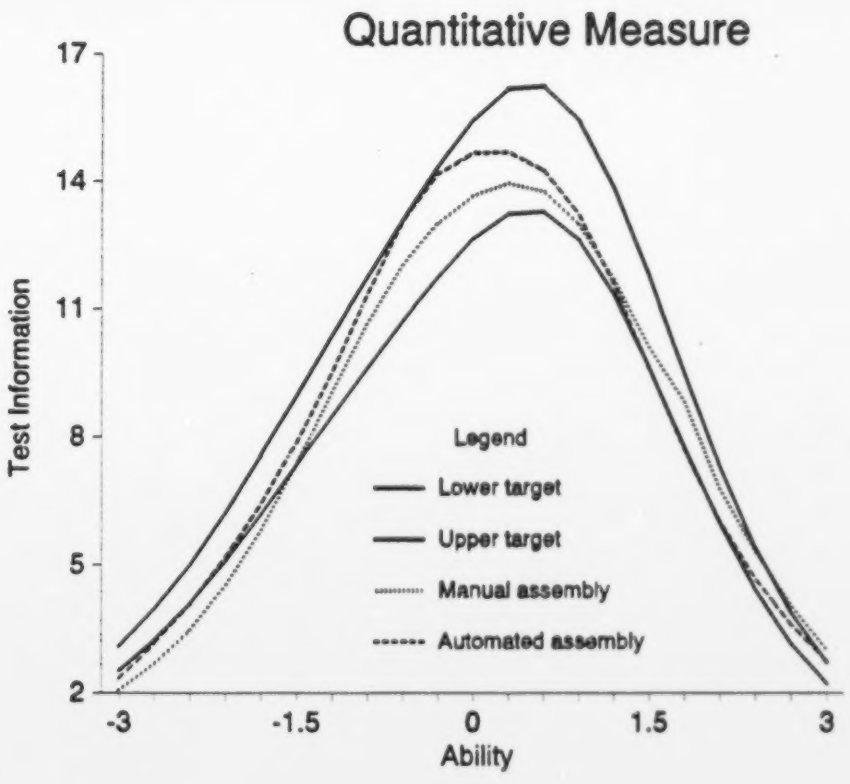

Figure 2: Upper and lower target test information functions and the resultant test information functions for the manual and automatic assemblies of the quantitative measure.